COMPUTER ENGINEERING DEPARTMENT

ASSIGNMENT NO-05

SUB: Computer Networks

COURSE: T.E. **Year: 2020-2021** **Semester: V**

DEPT: Computer Engineering

SUBJECT CODE: CSC503 **SUBMISSION DATE: 20/10/2020**

==========

Name: Amey Thakur **Roll No.:** 50

Batch: B3 **Class:** TE COMPS B

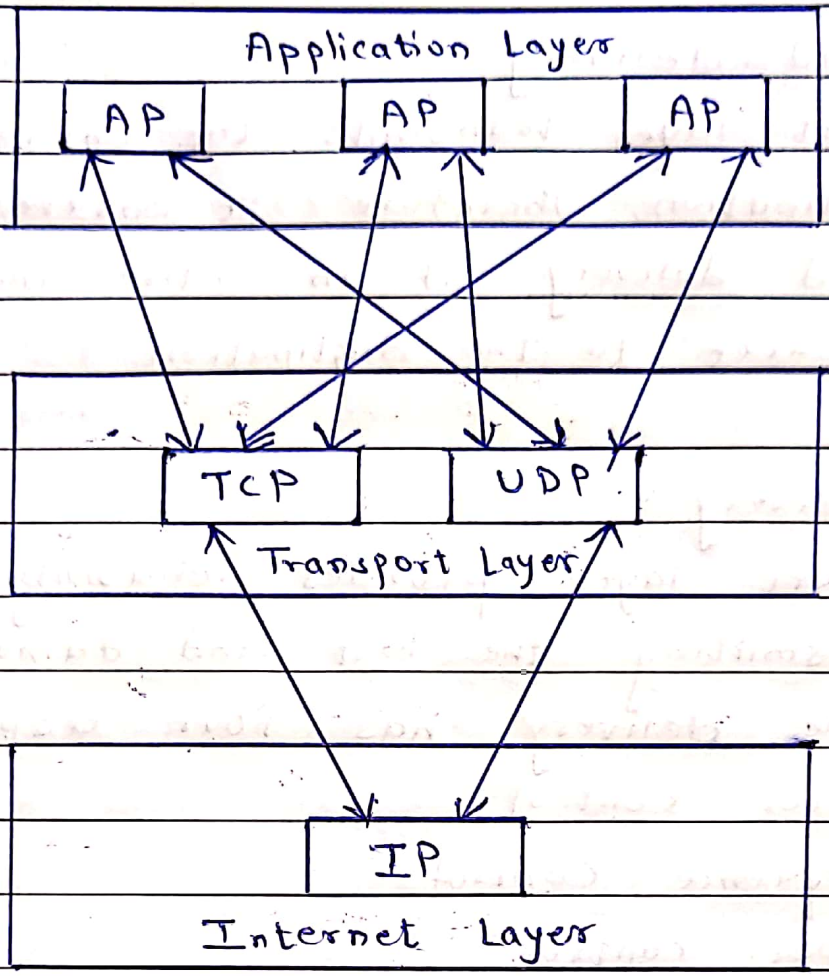
Questions to answer:

1. The Transport Services
2. Berkeley Sockets
3. UDP
4. TCP
5. Three-Way Handshake

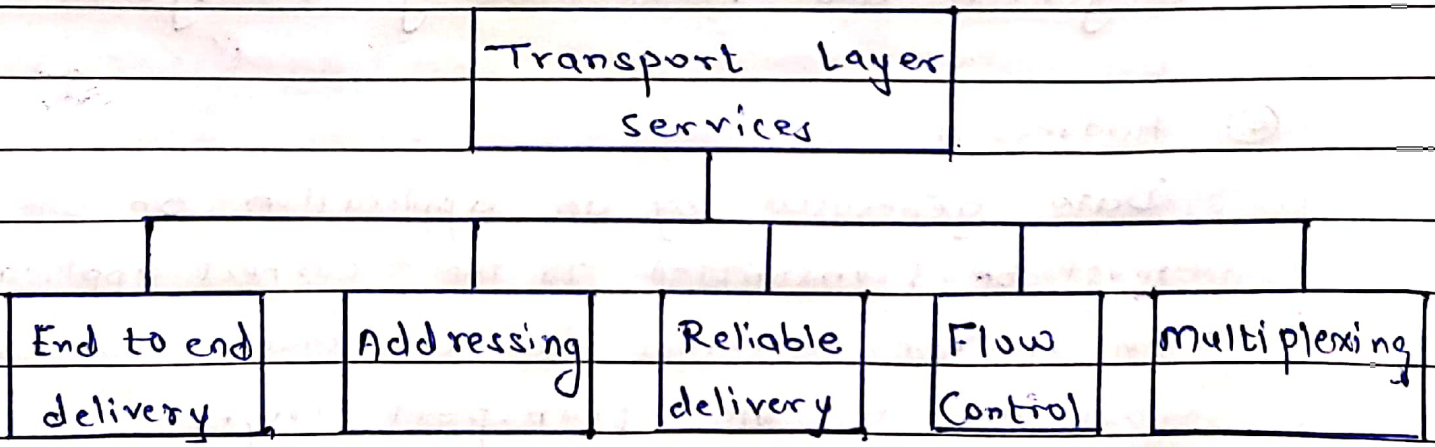
Q.1 The Transport Services

Ans:

- The transport layer is a 4th layer from the top.
- The main role of the transport layer is to provide the communication services directly to the application processes running on different hosts.
- The transport layer provides a logical communication between application processes running on different hosts. Although the application processes on different hosts are not physically connected, application processes use the logical communication provided by the transport layer to send the messages to each other.
- The transport layer protocols are implemented in the end systems but not in the network routers.
- A computer network provides more than one protocol to the network application.
  For example, TCP and UDP are two transport layer protocols that provide a different set of services to the network layer.
- All transport layer protocols provide multiplexing / demultiplexing service. It also provides other services such as reliable data transfer, bandwidth guarantees and delay guarantees.
- Each of the application in the application layer has the ability to send a message by using TCP or UDP. The application communicates using either of these two protocols. Both TCP and UDP will then communicate with the internet protocol in the internet layer.
  The application can read and write to the transport layer. Therefore, communication is a two way process



- The services provided by the transport layer protocols can be ~~divided~~ into five categories.
  1. End to end delivery
  2. Addressing
  3. Reliable delivery
  4. Flow control
  5. Multiplexing



① End to end delivery

- The transport layer transmits the entire message to the destination. Therefore, it ensures the end to end delivery of an entire message from a source to the destination.

② Reliable delivery

- The transport layer provides reliability services by retransmitting the lost and damaged packets.
- The reliable delivery has four aspects.
  1. Error control
  2. Sequence Control
  3. Loss control
  4. Duplicate Control

③ Flow control

- Flow control is used to prevent the sender from overwhelming the receiver. If the receiver is overloaded with too much data then the receiver discards the packets and asking for retransmission of packets. This increases network congestion and thus reducing the system performance.

④ Addressing

- Data generated by an application on one machine must be transmitted to the correct application on another machine. In this case, addressing is provided by the transport layer.

⑤ Multiplexing

- The transport layer uses the multiplexing to improve transmission efficiency.

Q.2. Berkeley Sockets.

Ans:

Socket:

- Sockets are a service provided by transport layer.
- A socket is one endpoint of a two way communication link between two programs running on the network.

Berkeley socket:

- Berkeley sockets is an application programming interface (API) for internet sockets and UNIX domain sockets.
- It is used for inter process communication (IPC).
- It is commonly implemented as a library of linkable modules.
- It originated with the 4.2 BSD UNIX released in 1983.

Primitive used in Berkeley Socket:

-

	Primitives	Meaning
①	SOCKET	Create a new communication endpoint.
②	BIND	Attach a local address to a SOCKET.
③	LISTEN	Shows the willingness to accept connections
④	ACCEPT	Block the caller until a connection attempt arrives
⑤	CONNECT	Actively attempt to establish a connection
⑥	SEND	Send some data over connection
⑦	RECEIVE	Receive some data from the connection
⑧	CLOSE	Release the connection.

Socket Programming:

① Server side:

- Server startup executes SOCKET, BIND and LISTEN primitives.
- LISTEN primitive allocate queue for multiple simultaneous clients.
- Then it use ACCEPT to suspend server until request.
- When client request arrives : ACCEPT returns.
- Start new socket (thread or process) with same properties as original, this handles the request, server goes on waiting on original socket.
- If new request arrives while spawning thread for this one, it is queued.
- If queue full it is refused.

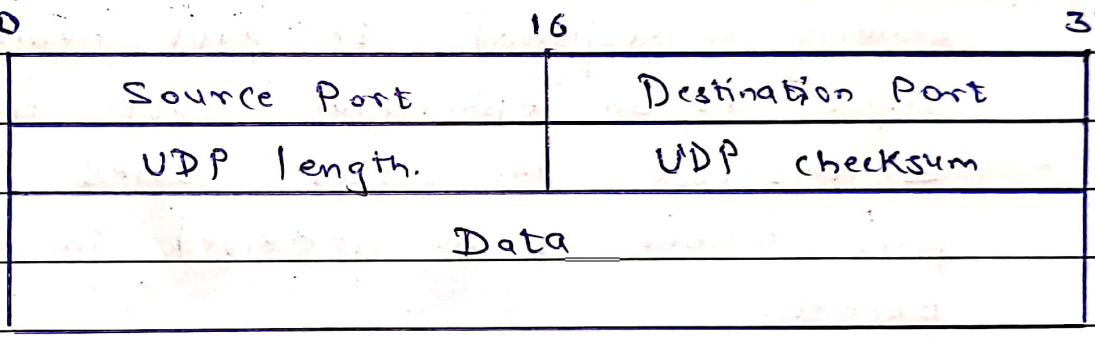
② Client side:

- It uses SOCKET primitives to create.
- Then use CONNECT to initiate connection process.
- When this returns the socket is open.
- Both side can now SEND, RECEIVE
- Connection not released until both sides do CLOSE.
- Typically client does it, server acknowledges.

Q.3. UDP

Ans:

- The User Datagram Protocol (UDP) is an unreliable, connectionless transport layer protocol. It is very simple protocol that provides only two additional services beyond IP : demultiplexing and error checking on data. UDP adds a mechanism that distinguishes among multiple applications in the host.
- UDP can optionally check the integrity of the entire UDP datagram. Applications that use UDP include Trivial File Transfer Protocol, DNS, SNMP, RIP.



Source Port number:

- This is the port number used by the process running on the source host.
- It is 16 bits long which means that the port number can range from 0 to 65,535.
- If the source host is the client (a client sending a request), the port number, in most cases is an ephemeral port number requested by the process and chosen by the UDP software running on the source host.
- If the source host is server (server sending a response) the port number, in most cases, is a well known port number.

Destination Port Number:

- This is the port number used by the process running on the destination host.
- It is also 16 bits long.
- If the destination host is a server (a client sending a request), the port number, in most cases, is a well known port number.
- If the destination host is a client (a server sending a response), the port number, in most cases, is an ephemeral port number.
- In this case, the server copies the ephemeral port number it has received in the request packet.

Length:

- This is a 16-bit field that defines the total length of the user datagram, header plus data.
- The 16 bit can define a total length of 0 to 65,535 bytes.

Checksum:

- This field is used to detect errors over the entire user datagram (header plus data)

Q.4. TCP

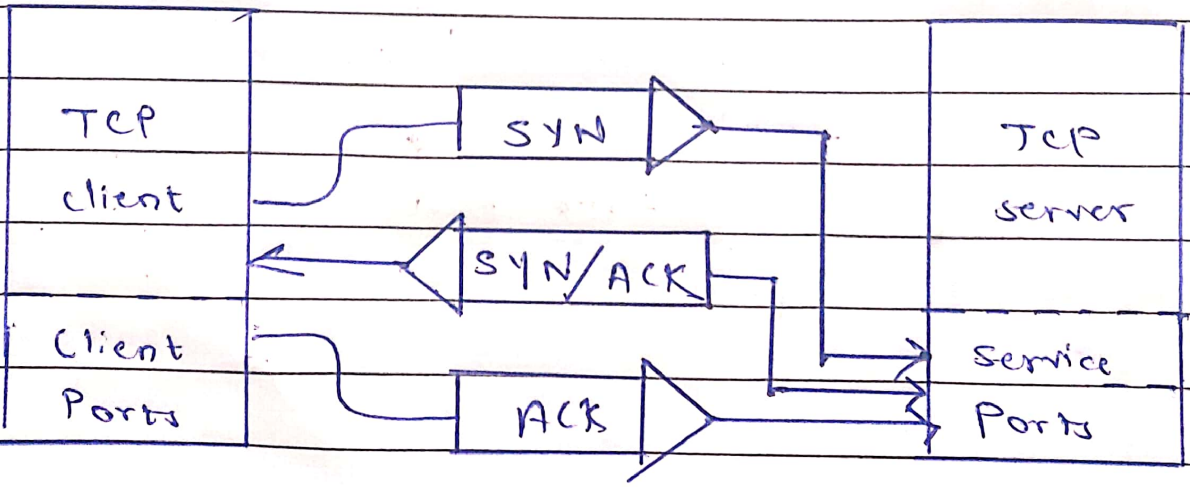
Ans:

- Transmission Control Protocol (TCP) is a connection oriented communications protocol that facilitates the exchange of messages between computing devices in a network.
- It is the most common protocol in networks that use the (IP) Internet Protocol. Together they are referred as TCP/IP.
- TCP takes messages from an application/server and divides them into packets which can then be forwarded by the device in the network - switches, routers, security gateways - to the destination.
- TCP numbers each packet and reassembles them prior to handling them off to the application/server recipient.
- Because it is connection oriented, it ensures a connection is established and maintained until the exchange between the application/servers sending and receiving the message is complete.

Q.5. Three way handshake

Ans:

- A three way handshake is a method used in a TCP/IP network to create a connection between a local host / client and server.
- It is a three step method designed to allow both communicating ends to initiate and negotiate the parameters of the network TCP socket connection at the same time before data such as HTTP and SSH is transmitted.
- Multiple TCP socket connection can be transmitted in both directions simultaneously.
- A three way handshake is also known as a TCP handshake or SYN - SYN - ACK and requires both the client and server to exchange SYN (synchronization) and ACK (Acknowledgement) packets before actual data communication begins.
- In fact, its name originates from the 3 message transmitted by TCP before the session between the two ends is initiated.



① Step 1 (SYN):

- In the first step, client wants to establish a connection with server so it ends a segment with SYN (Synchronized Sequence Number) which informs server that client is likely to start communication and with what sequence number it starts segment with.

② Step 2 (SYN + ACK):

- Server responds to the client request with SYN - ACK signal bits set.
- Acknowledgement (ACK) signifies the response it received and SYN signifies with what sequence number it is likely to start the segments with.

③ Step 3 (ACK):

- In the final part client acknowledges the response of server and they both establish a reliable connection with which they will start the actual data transfer.

- The steps 1, 2 establish the connection parameter for one direction and it is acknowledged.
- The steps 2, 3 establish the connection parameter for the other direction and it is acknowledged.
- With these, a full-duplex communication is established.

Note - Initial sequence numbers are randomly selected while establishing a connection between client and server.